

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и процессов управления

Кафедра Теории Управления

Сахаров Алексей Александрович

Выпускная квалификационная работа магистра

Детекция и распознавание средств регулирования железнодорожного
движения при помощи анализа видеопотока

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа ВМ.5517.2016

Методы прикладной математики и информатики в задачах управления

Научный руководитель, старший преподаватель

_____ Смирнов Михаил Николаевич

Рецензент:

_____ Пенкрат Николай Александрович

Санкт-Петербург

2018

РЕФЕРАТ

Отчет 47 страницы, 30 рисунков, 4 таблицы, 27 источников

Ключевые слова: компьютерное зрение, обработка изображений, метод Виолы-Джонса, контурный анализ, оператор Собеля, фильтр Калмана, OpenCV.

В работе рассматривается система диагностики актуальности расположения средств железнодорожного регулирования на примере знаков и светофоров при помощи анализа видеопотока. Рассмотрены существующие методы компьютерного зрения, приведен пример подобных систем для автомобильного транспорта. Предложен алгоритм работы исследуемой системы и выбраны методы анализа кадров видео. Обучен и протестирован каскадный классификатор Виолы-Джонса. Проведено сравнение четырех методов выделения границ. Рассмотрена применимость фильтра Калмана в задаче фильтрации шумов при выделении границ железнодорожных рельс. Описаны способы расчёта относительных координат найденного объекта. Предложены правила автоматизированного процесса выбора области интереса на основании информации, полученной при обработке предыдущего кадра.

ABSTRACT

47 pages, 30 pictures, 4 tables, 27 references.

Keywords: computer vision, image processing, Viola-Jones algorithm, edge detection, Sobel operator, Kalman filter, OpenCV.

The system of diagnostics of the actual location of the means of railway regulation is considered in the work using the example of signs and traffic lights with the help of analysis of the video stream. Existing methods of computer vision are considered, an example of similar systems for road transport is given. An algorithm for the operation of the system under investigation is proposed and methods for analyzing video frames are chosen. The Viola-Jones cascade classifier is trained and tested. Four methods of delimiting boundaries are compared. The applicability of the Kalman filter in the problem of filtering noise in the detection of railway rail boundaries is considered. The methods of calculating the relative coordinates of the found object are described. The rules of the automated process of selecting the area of interest based on the information received during the processing of the previous frame are proposed.

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ..... | 6 |
| ПОСТАНОВКА ЗАДАЧИ..... | 8 |
| ГЛАВА 1. ОБЗОР МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ И ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ ДЕТЕКЦИИ И КЛАССИФИКАЦИИ..... | 10 |
| 1.1. ОБЗОР МЕТОДОВ ВЫДЕЛЕНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ..... | 10 |
| 1.2. МЕТОД ВИОЛЫ-ДЖОНСА | 13 |
| 1.3. ОБРАБОТКА ИЗОБРАЖЕНИЯ НА ОСНОВЕ ИНФОРМАЦИИ О ЦВЕТЕ ОБЪЕКТА ИНТЕРЕСА | 14 |
| 1.4. АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ РАССТОЯНИЯ ДО ОБЪЕКТА..... | 14 |
| 1.5. ВЫДЕЛЕНИЕ ЛИНИЙ НА ИЗОБРАЖЕНИИ | 16 |
| 1.6. МЕТОД ХАФА | 17 |
| 1.7. ФИЛЬТР КАЛМАНА | 19 |
| ГЛАВА 2. ОПИСАНИЕ ПРОЦЕДУРЫ ИССЛЕДОВАНИЯ..... | 21 |
| 2.1. ПОСТРОЕНИЕ ВЫБОРКИ ИЗОБРАЖЕНИЙ ДЛЯ ОБУЧЕНИЯ И ТЕСТИРОВАНИЯ КЛАССИФИКАТОРА | 21 |
| 2.2. ОБУЧЕНИЕ КАСКАДНОГО КЛАССИФИКАТОРА ВИОЛЫ-ДЖОНСА | 22 |
| 2.3. ДЕТЕКТИРОВАНИЕ ЖЕЛЕЗНОДОРОЖНОЙ КОЛЕИ ДЛЯ ПОСТРОЕНИЯ ПРОЕКЦИИ ИЗОБРАЖЕНИЯ НА ТРЕХМЕРНОЕ ПРОСТРАНСТВО..... | 25 |
| 2.3.1. Метод Кенни и его адаптация для выделения контуров на изображении | 27 |
| 2.3.2. Применение метода Хафа и интерпретация его результатов... | 27 |
| 2.3.3. Виды ошибок при выделении рельс. Применение фильтра Калмана | 29 |
| 2.4. ПОСТРОЕНИЕ ПЛОСКОСТИ ПО ДВУМ ПАРАЛЛЕЛЬНЫМ ЛИНИЯМ И ОПРЕДЕЛЕНИЕ КООРДИНАТ ОБЪЕКТА ИНТЕРЕСА | 30 |
| 2.5. ВЫБОР ОБЛАСТИ ИНТЕРЕСА В СООТВЕТСТВИЕ С ФОРМОЙ ЖЕЛЕЗНОДОРОЖНОЙ КОЛЕИ | 33 |

| | |
|--|----|
| 2.6. ВЫВОД ПО ГЛАВЕ..... | 36 |
| ГЛАВА 3. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ | 38 |
| 3.1. КОНТРОЛЬ ТОЧНОСТИ КЛАССИФИКАЦИИ | 38 |
| 3.2. РЕЗУЛЬТАТЫ СРАВНЕНИЯ МЕТОДОВ ВЫДЕЛЕНИЯ КОНТУРА РЕЛЬС | 40 |
| 3.3. КОНТРОЛЬ СКОРОСТИ ОБРАБОТКИ ОДНОГО КАДРА ВИДЕО | 43 |
| ВЫВОДЫ..... | 45 |
| СПИСОК ЛИТЕРАТУРЫ | 46 |

ВВЕДЕНИЕ

В последние годы приобретают большую популярность цифровая обработка и цифровой анализ изображений. Они применяются в различных прикладных областях. От автоматизированной медицинской диагностики до беспилотного управления автомобилем, от глубоководных аппаратов, до посвящено искусственных космических спутников. Возможность применения «компьютерного зрения» многогранна, а решение задач в одной сфере может быть перенесено на совершенно не связанную с ней прикладную область. Обработке образов не малое количество работ, однако «универсального» решения для любой задачи на данный момент не создано. Каждая прикладная область накладывает свои специфичные условия. Исходя из данных условий необходимо построить алгоритм, который решает задачу с максимальной быстротой и точностью, но как правило, остаются условия, которые преодолеть гораздо сложнее, чем другие. К примеру, большинство методов не устойчиво к смене интенсивности или направления освещения, в то время, как другие плохо справляются с изменением масштаба и т.п.

В рамках данной работы в качестве прикладной области рассматривается автоматизированный анализ видеопотока с камеры из кабины машиниста поезда.

Уровень экономики страны напрямую связан с её техническим развитием. Инновационные технологии помогают в решении различных практических задач. В то время, как цели этих задач могут быть совершенно различными: увеличение прибыли от экономических процессов, ускорение доставки товаров, увеличение прочности изделий и т.д. Одной из таких целей является увеличение безопасности процесса.

Нельзя не отметить высокую значимость транспортного вопроса при решении экономической задачи. С ростом требований к скорости движения возникает и необходимость повышения его безопасности. На протяжении многих лет железнодорожный транспорт продолжает активно использоваться, как один из основных видов транспорта в нашей стране. В виду её большой

протяженности с запада на восток, в большинстве районов он становится более эффективным, чем автотранспорт. Для обеспечения его активного использования и развития необходимо внедрять принципиально новые технические решения, которые снижают возможные негативные факторы, подвергающие движение риску. В качестве таких факторов можно выделить неблагоприятные погодные условия, человеческий фактор машиниста электропоезда, окружающая инфраструктура и т.д. В связи с этим можно выделить необходимость технического содействия машинисту в принятии решений по управлению поездом.

На железной дороге, как и автомобильной, присутствуют строго определенные правила движения, руководствуясь которыми машинист предпринимает действия различного рода. Для указания на данные действия вдоль железной дороги располагаются регулирующие устройства: знаки, световые устройства, устройства подачи звукового сигнала [5], автоматическая локомотивная сигнализация (АЛС) [3] – система сигнализации, передающая сигналы по рельсовой линии на локомотив. Корректное их расположение и состояние дает большой объем информации. В связи с большой протяженностью участков железной, к которым нет прямого доступа, кроме непосредственного проезда по ним возникает необходимость в локомотивной системе контроля расположения данных регулирующих устройств. На корректности расположения регулирующих средств могут повлиять как неблагоприятные погодные условия, так и человеческий фактор (вандализм). Отсутствие необходимого знака может привести к ошибке машиниста, а опора только на геоинформационные системы невозможна.

Задача детекции и распознавания дорожных знаков имеет различные способы решения в сфере автомобильного транспорта, в то время как, сходная задача для железнодорожного транспорта описана в достаточно малом количестве работ.

ПОСТАНОВКА ЗАДАЧИ

Постановку задачи для данной работы можно сформулировать следующим образом: Разработать алгоритм системы детекции, распознавания железнодорожных знаков. В случае успешного решения, это позволит в связке с GPS оборудованием успешно решать вопрос геолокации и точного сравнения факта с плановым расположением знака.

Для построения классификатора необходимо наличие достаточного набора обучающих и тестовых изображений. К сожалению, для железнодорожного транспорта отсутствуют подходящие наборы изображений, что приводит к необходимости самостоятельно собирать примеры изображений. В связи с этим, в рамках данной работы так же рассмотрен вопрос построения и применение обучающего набора изображений.

Актуальность: в отличие от автодорог, где уже достаточно давно и активно развиваются разнообразные системы содействия водителю (ADAS) [25, 27], для железнодорожного транспорта системы анализа путевой инфраструктуры перед локомотивом почти не представлены. В то время как масса, а, следовательно, и тормозной путь, поезда во много раз больше, чем у автотранспорта. В некоторых случаях он составляет несколько километров. Система, позволяющая контролировать актуальность регулирующего движение инфраструктуры железной дороги может значительно обезопасить движение и защитить от чрезвычайных ситуаций [6].

Цель работы: разработка и программная реализация алгоритма, проверяющего актуальность существующей регулирующей железнодорожной инфраструктуры.

Объект исследования: локомотивная система анализа актуальности регулирующей информационной инфраструктуры.

Предмет исследования: разработка локомотивной аналитической системы анализа видеoinформации.

Задачи исследования:

1. Рассмотреть алгоритмы выделения объектов на изображении.
2. Составить обучающий и тестовый наборы изображений для построения алгоритма выделения объектов интереса.
3. Обучить и протестировать алгоритм выделения объектов на изображении.
4. Рассмотреть алгоритмы определения расстояния до объекта на изображении.
5. Рассмотреть вопрос оптимизации выбора области интереса, путем автоматизированного анализа расположения железнодорожной колеи.

Для решения поставленных в данной работе задач были использованы методы цифровой обработки изображений, теория распознавания образов, применяется вычислительная техника и среда программирования Microsoft Visual Studio с использованием функций библиотеки OpenCV (EmguCV C#).

ГЛАВА 1. ОБЗОР МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ И ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ ДЕТЕКЦИИ И КЛАССИФИКАЦИИ

1.1. Обзор методов выделения объектов на изображении

В настоящий момент разработано большое количество алгоритмов поиска объекта интереса на изображении. Как правило, невозможно построить алгоритма, способного покрыть все возможные классы объектов. В связи с этим, обзор существующий методов является одним из первых этапов решения задачи обработки изображений.

В статье [17] рассмотрены наиболее популярные группы алгоритмов выделения информации на изображении, описаны их преимущества, недостатки и область применения. Выделено 3 группы алгоритмов:

- Color-based (выделение области интереса на основании цвета);
- Shape-based (основным критерием является анализ формы и контура объекта);
- Learn-based (машинное обучение – основной инструмент анализа).

В вопросах детекции и анализа расположения дорожных знаков достаточно часто находит своё применение подход, основанный на поиске цвета интереса. Как правило, информация о знаке закреплена в различных правовых актах, в связи с этим можно быть уверенным, что знак всегда сохраняет свои цвета и форму. Однако, алгоритмы данной группы подвержены высокому влиянию изменения освещенности объекта, из-за чего требуется применение динамически определяемых параметров. Данные алгоритмы становятся существенно устойчивее при переходе от RGB цветовой схемы (Red, Green and Blue) в пространство HSV (Hue, Saturation and Value). В данной цветовой схеме можно задать жесткие ограничения на тон интереса (hue), и различные пороги насыщенности (Saturation) и яркости (Value) цвета.

Подробное описание цветовых пространств и методов перехода между ними приведено в книге [1].



Рисунок 1 Результат выделения синей компоненты на изображении

Алгоритмы данной группы применимы в задаче предварительного выделения области с повышенной вероятностью нахождения объекта интереса. Результаты можно передавать на следующий этап в качестве гипотезы о его местоположении.

Анализ формы объекта основан на вычислении градиента изменения цвета для каждого пикселя. Далее вычисленные градиенты анализируются с предположением о том, что объект интереса может быть описан некоторым регулярным правилом. К примеру, для поиска треугольника можно утверждать, что градиент будет направлен по нормали к стороне данного многоугольника. При этом нормали к сторонам треугольника подчиняются соотношению, приведенному на рисунке 2:

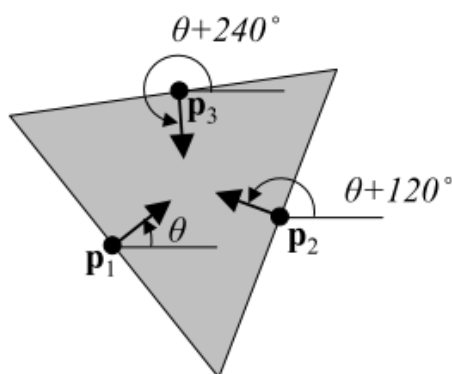


Рисунок 2 Соотношение направление нормалей в равностороннем треугольнике

Это означает, что чем больше групп из трех таких нормалей удалось выделить в ограниченной области, тем выше вероятность нахождения там рассматриваемого объекта.

Вероятно, наиболее популярным алгоритмом данной группы является преобразование Хафа [14] и его обобщение, предложенное в 1981 году D.H. Ballard [9]. Классический алгоритм преобразования Хафа связан с идентификацией прямых в изображении, но позже алгоритм был расширен возможностью идентификации позиции произвольной фигуры, чаще всего эллипсов и окружностей.

В подходах, описанных выше, основные знания об объекте интереса заранее закладываются в алгоритм обнаружения. Однако, данная информация может быть получена с помощью машинного обучения.

Исследования Виолы и Джонса [24] продемонстрировали значительную веху в компьютерном зрении. Виола и Джонс разработали алгоритм, позволяющий обнаруживать объекты очень надежно и в режиме реального времени. Данный алгоритм использует примитивы Хаара, для определения вероятности нахождения объекта в данной области. Использование интегрального представления изображения позволяет сильно ускорить процесс вычисления примитивов Хаара, простые признаки усиливаются при помощи технологии AdaBoost (Adaptive Boosting), а каскадная архитектура классификатора позволяет быстро отбрасывать окна, в которых отсутствует объект интереса.

Также нельзя не отметить активное применение нейронных сетей в области компьютерного зрения. К примеру, в задаче детекции и распознавания дорожных знаков часто применяется свёрточная архитектура нейронных сетей, предложенная Яном Лекуном в 1988 году [18]. Идея данных многослойных сетей основана на выполнении многократной операции свертки между изображением и ядром свертки. Они предназначены для выделения абстрактных понятий изображения, при этом сами эти понятия выстраиваются в процессе обучения сети.

Нейронные сети являются мощным инструментом в области компьютерного зрения. В качестве объекта интереса могут выступать различные классы предметов. Однако нейронные сети, как правило требуют

больших вычислительных затрат, особенно на этапе обучения. В связи с этим, в данной работе было решено использовать метод Виолы-Джонса.

1.2. Метод Виолы-Джонса

Метод Виолы-Джонса – алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени. Предложен в 2001 году Paul Viola и Michael Jones [24]. Этот детектор обладает высокой скоростью обнаружения и крайне низкой вероятностью ложного срабатывания. Алгоритм хорошо работает и распознает под небольшим углом, примерно до 30 градусов. При угле наклона больше 30 градусов процент обнаружений резко падает [11].

Основные принципы, на которых основан метод:

- используются признаки Хаара [24], с помощью которых происходит поиск нужного объекта;
- используются интегральное представление изображения, что позволяет быстро вычислять быстро необходимые признаки;
- отдельные признаки Хаара объединяются в единый алгоритм при помощи технологии бустинга (от англ. boost – улучшение, усиление) для выбора, наиболее подходящего на данном участке изображения признака;
- используются каскады признаков для быстрого отбрасывания окон, где не найден объект.

Данный алгоритм не теряет своей популярности. К примеру, в 2002 году был предложен расширенный набор признаков Хаара [19]. Реализация каскадного классификатора в библиотеке OpenCV содержит, как базовые признаки, так и расширенный набор. В работе [8] 2017 года рассматриваются различные алгоритмические улучшения подхода Виолы-Джонса.

1.3. Обработка изображения на основе информации о цвете объекта интереса

Классификатор Виолы-Джонса предназначен для работы с изображениями в оттенках серого. Данное ограничение может приводить к ложным срабатываниям в случае повторяющихся форм объектов.

Для уменьшения вероятности ложных срабатываний можно применить пороговую фильтрацию в цветовых пространствах HSL (Hue-Saturation-Lightness, тон-насыщенность-светлота) или HSV (Hue-Saturation-Value, тон-насыщенность-яркость). Отображение данных цветовых пространств приведено на рисунке 3:

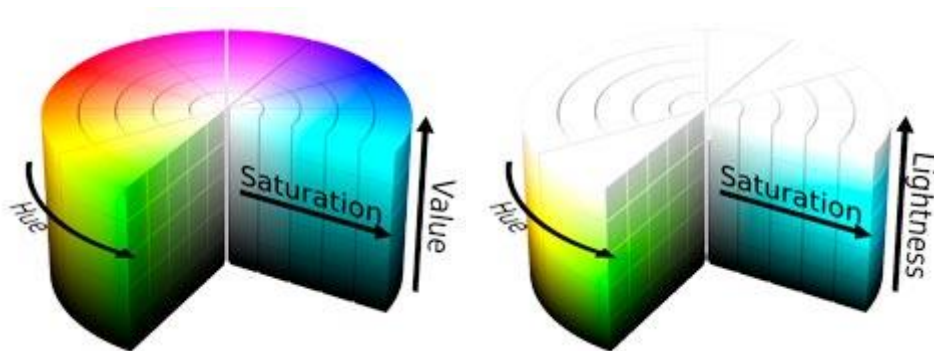


Рисунок 3 Представление цветовых пространств HSV и HSL в форме цилиндра

К примеру, для выделения области желтого цвета можно ограничить только значение показатель тона, что сделает данный пороговый фильтр менее чувствительным к освещенности изображения. А при поиске областей белого или черного цвета необходимо ограничить уровень светлоты изображения.

1.4. Алгоритмы определения расстояния до объекта

Т.к. в рамках данной работы кроме вопроса обнаружения средств железнодорожного регулирования рассматривается задача валидации их местоположения, то кроме поиска объекта интереса на снимке необходимо определить его расположение относительно камеры.

В общем случае информации на фото снимке (видео кадре) не достаточно для вычисления расстояний до объектов. В связи с этим существует несколько подходов к решению данной задачи.

- Использование двух камер.

Пример такой системы можно найти в [7]. Основная идея данного подхода заключается в том, чтобы поставить в соответствие точки двух снимков с разных камер и по известным углам направления оси камер и расстоянию между камерами произвести расчет расстояния до объекта. Данный подход предполагает, что камеры находятся на достаточном удалении. Это необходимо для минимизации влияния погрешности определений углов на объект. Также стоит отметить, что использование двух камер увеличивает стоимость системы. К плюсам данной системы можно отнести возможность вычисления расстояний сразу до нескольких объектов в один момент времени.

- Видео камера и лазерный дальномер.

При использовании данного подхода камера используется в качестве устройства определения углов на точку интереса. Направление на данную точку передается на дальномер, и с его помощью вычисляется расстояние до данной точки. При применении данного подхода возникает необходимость регулярной настройки дальномера на точку интереса. Это приводит к невозможности параллельных измерений нескольких объектов и существенно снижает скорость анализа. Однако, данный подход является достаточно точным и используется при необходимости проведения точных расчетов расстояния до единичных объектов.

- Одна камера и объект с известными размерами.

В данном случае объект с известными размерами (контрольный объект) должен находиться как можно ближе к объекту интереса. На изображении камеры вычисляется размер контрольного объекта в пикселях. Далее расстояние можно вычислить по формуле:

$$L = \frac{D * k}{M},$$

где k - постоянный коэффициент (зависит от характеристик камеры), M – в пикселях, D - реальный размер объекта.

В контексте данной работы был выбран третий подход, т.к. он является менее дорогим в реализации. В то же время, железнодорожная колея может быть применена в качестве объекта с известными параметрами.

1.5. Выделение линий на изображении

Для выделения железнодорожной колеи воспользуемся методом Кенни [11] с оператором Собеля [23] для детекции границ и обобщенным методом Хафа [14, 9] для выделения прямых линий.

Метод Кенни выделения границ реализует следующие этапыЖ

1. Сглаживание. К изображению применяется размытие Гаусса для удаления шума.
2. Поиск градиентов. В каждой точке изображение определяется направление градиента и его значение по модулю. В качестве границы выбираются пиксели с максимальным значением модуля градиента.
3. Подавление немаксимумов. Данный термин был введен Кенни в работе [11]. Он обозначает, что только локальные максимумы могут быть объявлены контуром на изображении.
4. Двойная пороговая фильтрация. При фильтрации немаксимумов используется два значения порогов. Первый предназначен для уменьшения влияния шумов, второй для сохранения непрерывности границ в случае неравномерного распределения градиента.

Приближенное вычисление градиентов может производиться при помощи оператора Собеля. Для вычисления компонент по осям x и y производится свертка с двумя операторами:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A$$

Далее модуль и направление градиента вычисляются по следующим формулам:

$$G = \sqrt{G_x^2 + G_y^2}, \theta = \arctan\left(\frac{G_x}{G_y}\right)$$

На следующем рисунке приведен пример выделения границ описанным алгоритмом. а – начальное изображение; б – результат свертки с оператором G_x ; в – результат свертки с оператором G_y ; г – результирующее изображение после обработки пороговым фильтром.

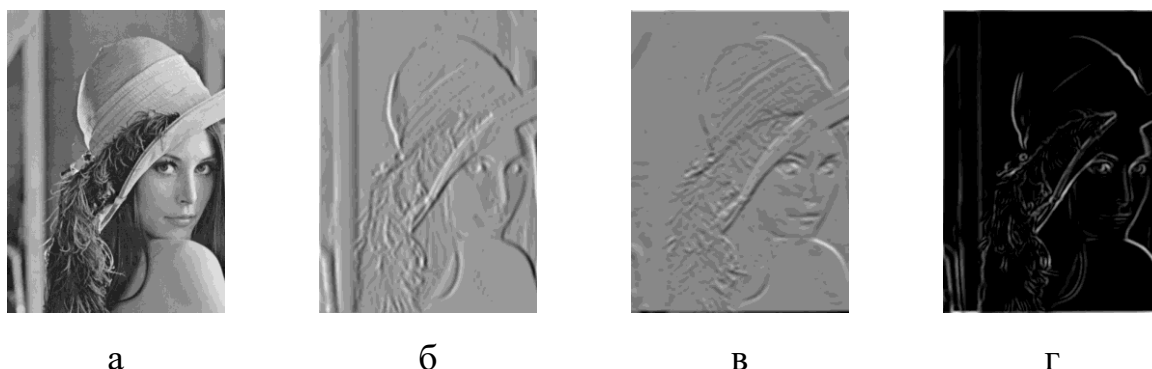


Рисунок 4 Визуализация процедуры выделения границ при помощи операторов Собеля

1.6. Метод Хафа

Преобразование Хафа [14, 9], — алгоритм, применяемый для извлечения элементов из изображения, и предназначен для поиска объектов, принадлежащих определённому классу фигур с использованием процедуры голосования. Процедура голосования применяется к пространству параметров, из которого и получают объекты определённого класса фигур по локальному максимуму в, так называемом, накопительном пространстве (accumulator space), которое строится при вычислении трансформации Хафа.

В простейшем случае преобразование Хафа является линейным преобразованием для обнаружения прямых. Прямая может быть задана с помощью других параметров, известных как ρ и θ (ρ, θ). Параметр ρ — это длина радиус-вектора ближайшей к началу координат точки на прямой, а θ — это угол между этим вектором и осью абсцисс.

Таким образом, уравнение прямой можно записать как:

$$y = \left(-\frac{\cos \theta}{\sin \theta}\right)x + \left(\frac{\rho}{\sin \theta}\right)$$

или после преобразования:

$$\rho = x \cos \theta + y \sin \theta$$

Поэтому возможно связать с каждой прямой на исходном изображении (в плоскости X-Y) точку с координатами (ρ, θ) , где $\theta \in [0; \pi]$ и $\rho \in \mathbf{R}$, или что где $\theta \in [0; 2\pi]$ и $\rho > 0$.

Плоскость (ρ, θ) иногда называется Пространством Хафа для множества прямых в 2-мерном случае. Через каждую точку плоскости может проходить бесконечно много прямых. Если эта точка имеет координаты (x_0, y_0) , то все прямые, проходящие через неё, соответствуют уравнению:

$$\rho(\theta) = x_0 \cos \theta + y_0 \sin \theta$$

Что соответствует синусоидальной линии в пространстве Хафа (ρ, θ) . Если эти синусоиды пересекаются, то данная точка (в пространстве Хафа), соответствует прямым (в оригинальном месте изображения), которые проходят через обе точки, т.е. совпадают. Таким образом, проблема обнаружения коллинеарных точек может быть сведена к проблеме обнаружения пересекающихся кривых.

Визуальное представление преобразование Хафа приведено на рисунке 5. Точки пространства параметров с наибольшим количеством пересечений принимаются в качестве характеристических точек исследуемых прямых.

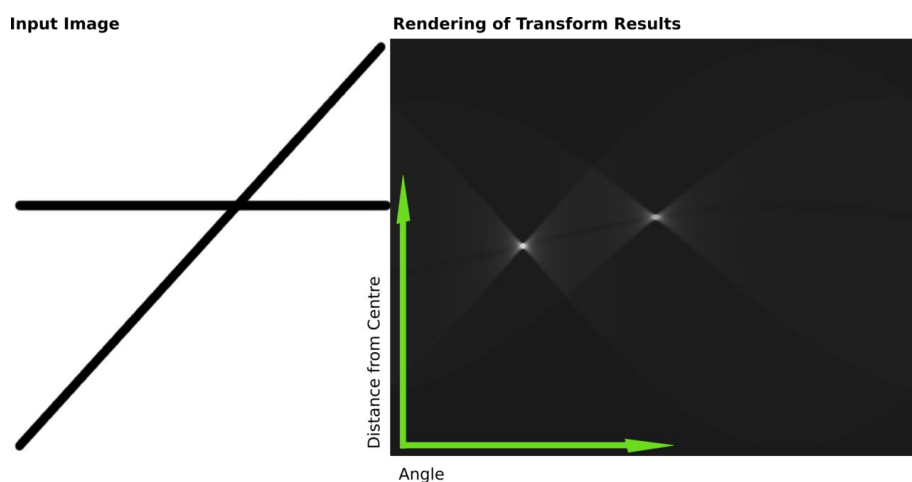


Рисунок 5 Визуальная интерпретация преобразования Хафа

1.7. Фильтр Калмана

Фильтр Калмана – рекурсивный фильтр, оценивающий вектор состояния динамической системы на основе зашумленных данных [15, 2].

В ситуации работы с техническими системами возникает вопрос корректного оценивания вектора состояния системы. Задачу наблюдения динамической системы записывается следующим образом:

$$\begin{cases} \dot{x}(t) = A(t) * x(t) + B(t)u(t), t_0 \leq t \\ y(t) = C(t)x(t) \end{cases} \quad (1)$$

Где $x(t)$ – n -мерный неизвестный вектор состояния системы, $u(t)$ – p -мерный известный вектор управления и y – вектор выходных величин. Требуется определить оценку $\tilde{x}(t)$.

Рассмотрим 2 случайных величин $\xi(t)$ – внутренние возмущения системы и $\eta(t)$ – погрешность измерения. В таком случае система (1) может быть приведена к виду:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) + \xi(t), t_0 \leq t \\ y(t) = C(t)x(t) + \eta(t) \end{cases} \quad (2.1)$$

Или в дискретном случае:

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + \xi_k \\ y_k = C_k x_k + \eta_k \end{cases} \quad (2.2)$$

При этом:

- u_k — известная переменная шага k ;
- ξ_k, η_k — случайные величины. И их законы распределения не зависят номера шага k ;
- Математические ожидания ошибок равны: $E\xi_k = E\eta_k = 0$;
- Как правило, известны их дисперсии: σ_ξ^2 и σ_η^2 ;
- Предполагается, что все случайные ошибки независимы друг от друга.

Для вычисления оценки вектора состояния применяется следующая формула:

$$\tilde{x}_{k+1} = K_{k+1} C_k y_{k+1} + (1 - K_{k+1})(A_k \tilde{x}_{k+1} + B_k u_k) \quad (3)$$

Коэффициент \mathbf{K} называют коэффициентом Калмана. Идея фильтрации состоит в том, чтобы подобрать коэффициент таким образом, что ошибка:

$$e_{k+1} = x_{k+1} - \tilde{x}_{k+1}$$

достигает своего минимального значения. Т.е. $E(e_{k+1}^2) \rightarrow \min$.

При $A_k = B_k = C_k = \mathbf{E}$ можно показать, что:

$$E(e_{k+1}^2) = (1 - \mathbf{K}_{k+1})^2 (Ee_k^2 + \sigma_\xi^2) + \mathbf{K}_{k+1}^2 \sigma_\eta^2$$

Минимум данной функции достигается при \mathbf{K} :

$$\mathbf{K}_{k+1} = \frac{Ee_k^2 + \sigma_\xi^2}{Ee_k^2 + \sigma_\xi^2 + \sigma_\eta^2}$$

Среднее значение квадрата ошибки может быть записано в виде:

$$E(e_{k+1}^2) = \frac{\sigma_\eta^2 (Ee_k^2 + \sigma_\xi^2)}{Ee_k^2 + \sigma_\xi^2 + \sigma_\eta^2} \quad (4), \text{ при этом } \mathbf{K}_{k+1} = \frac{E(e_{k+1}^2)}{\sigma_\eta^2} \quad (5)$$

Формулы 4 позволяют итерационно вычислять среднее значение квадрата ошибки и коэффициент Калмана на шаге k при известных дисперсиях ошибок системы. Формула 3 позволяет вычислять приближенную оценку вектора состояния системы.

ГЛАВА 2. ОПИСАНИЕ ПРОЦЕДУРЫ ИССЛЕДОВАНИЯ

2.1. Построение выборки изображений для обучения и тестирования классификатора

В задаче машинного обучения качестве начальных данных рассматриваются два набора изображений, снабженных дополнительной информацией. Первый набор предназначен для обучения классификатора, второй – для тестирования его работы в целях оценки и оптимизации применяемого алгоритма.

Для определения структуры начальных данных можно обратиться к аналогичной задаче на автомобильной дороге. Для тестирования классификаторов систем автоматического содействия водителю (ADAS) [25, 27] применяются различные наборы изображений. Одним из наиболее востребованных таких наборов является «KITTI Object Detection Evaluation 2012» [13]. На данный момент он содержит 7481 обучающее и 7518 тестовых изображений. На этих изображениях в общей сложности выделено более 80 000 объектов различной природы. Для объекта указаны данные его расположения и оценка ориентации и высоты относительно дороги. «CVCL StreetScenes Challenge Framework» [10] содержит более скромный набор изображений: около 3500. На них выделено около 7500 объектов. К сожалению, не удалось найти подобных аналогов, поэтому было решено создать собственный набор данных для обучения классификатора.

На железнодорожном транспорте создать подобную обучающую выборку в короткие сроки достаточно сложно. Во-первых, по причине отсутствия достаточного количества начальных, неразмеченных изображений. Во-вторых, закрытость организаций, осуществляющих железнодорожные перевозки не позволяет допустить постороннего человека к ведению фото-, видеосъемки из кабины машиниста, в то время как ракурс съемки имеет определяющее значение, так как классификатор, обученный для работы под

одним ракурсом съемки может оказаться полностью неприменим для обработки изображений, снятых под другим ракурсом.

Так как построение набора начальных данных на основе реальной фото-, видеосъемки трудновыполнимо, то стоит обратиться к синтезированным и виртуальным данным. Так в статье «Video Games for Autonomous Driving» [12] рассматривается интересный подход сбора изображений: используются записи видеоигры Grand Theft Auto 5 (GTA 5). Данная видеоигра достаточно хорошо отвечает реальностям современного мира, дороги и дорожная инфраструктура проработаны с особым вниманием, а большой и разнообразный игровой мир позволяет найти интересующие объекты.

Подобная способ построения начальной выборки существенно упрощает процесс работы над алгоритмом и позволяет создать демонстрационный образец без применения усилий по сбору изображений реального мира.

В рамках данной работы для сбора изображений был применен симулятор железнодорожного транспорта Train Simulator [26], разработанный компанией Dovetail Games.

2.2. Обучение каскадного классификатора Виолы-Джонса

Железнодорожные средств регулирования движения представлены различными типами знаков, светофором и семафоров. В качестве объектов интереса в данной работе рассмотрены следующие три вида:

- Знак ограничения скорости движения (жёлтый круг):



Рисунок 6 Знак ограничения скорости движения

- Знаки приближения к железнодорожному переезду.

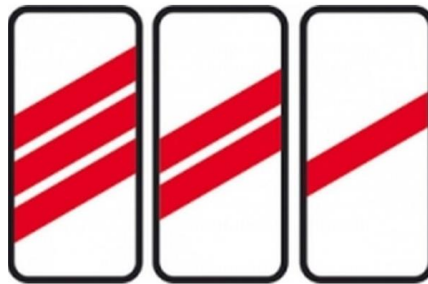


Рисунок 7 Знак приближения к ж/д переезду

- Железнодорожный светофор немецкой транспортной сети типа Н\V (был выбран в связи с тем, что симулятор не поддерживает железнодорожные светофоры, применяемые в России).



Рисунок 8 Железнодорожный светофор

Все три средства регулирования были выбраны из разных категорий с целью исследования различных способов детектирования. Для каждого из них была подготовлена обучающая выборка и построен каскадный классификатор Виолы-Джонса.

Обучающая выборка включает:

- 1074 изображений, не содержащий объектов интереса;

- 525 изображений знака ограничения скорости;



Рисунок 9 Пример изображений, использованных в процессе обучения

- 420 изображений знака железнодорожного переезда;



Рисунок 10 Пример изображений, использованных в процессе обучения

- 327 изображений светофора.

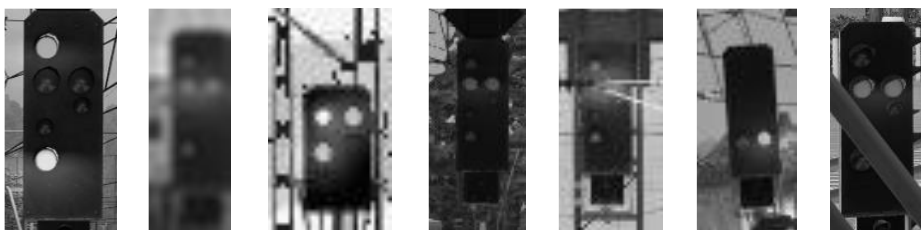


Рисунок 11 Пример изображений, использованных в процессе обучения

Для увеличения объема обучающей выборки к начальным изображениям применялись преобразования вида:

- Добавление гауссовского шума;
- Добавление размытия изображения.

Симулятор позволяет моделировать различные погодные условия, изменять время суток и сезон года. В обучающую выборку вошли изображения, сделанные при дневном и вечернем освещении; применялись эффекты дождя и снега. Ночные снимки были исключены из рассмотрения.



Рисунок 12 Пример изображений, содержащего эффект снега

Поскольку типы объектов интереса различны, а метод Виолы-Джонса предназначен для выделения общих особенностей, то было решено обучить три классификатора, каждый из которых отвечает за свой объект интереса. В процессе обучения использовалась библиотека OpenCV и приложения `opencv_createsamples`, `opencv_traincascade` and `opencv_visualisation` [21].

2.3. Детектирование железнодорожной колеи для построения проекции изображения на трехмерное пространство

В параграфе 1.4 были рассмотрены различные способы определения расстояния до объекта на изображении. Т.к. целью данной работы является проверка расположения средств железнодорожного регулирования, то возникает необходимость построения проекции плоскости рисунка на трехмерное пространство. Данную проекцию было решено строить при помощи определения плоскости железнодорожной колеи и расстояния до неё в различных точках изображения.

Варианты ширины колеи и их распространение по странам приведены на рисунках 13, 14.

Вводится несколько предположений:

- Железнодорожная колея непрерывна на расстоянии достаточном для построения плоскости.

- Предполагается постоянная ширина колеи. В структуре железнодорожной колеи возможна механическая деформация. Так в статье [16] рассматривается анализ механических искривлений рельс в процессе эксплуатации.
- Искривление железнодорожной колеи на участках маневра не выходит за пределы области изображения.
- Искривление железнодорожной колеи на участках маневра не приводит к ситуации, при которой оба рельса расположены левее или правее оси камеры.

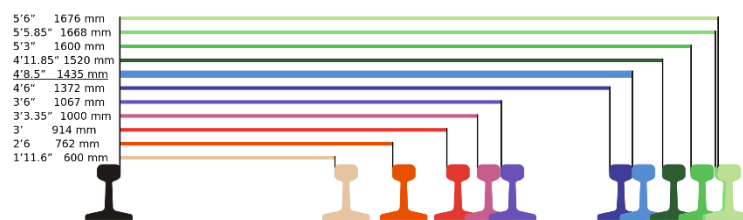


Рисунок 13 Ширина ж/д колеи

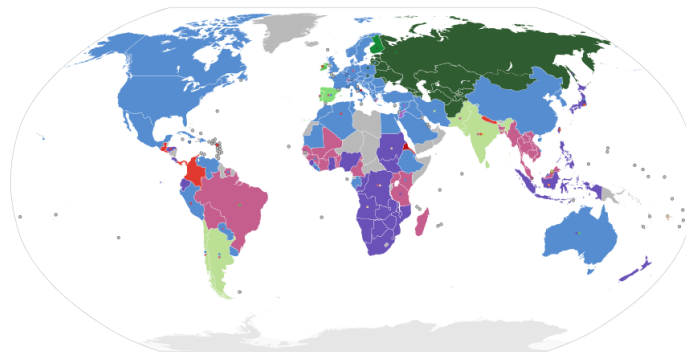


Рисунок 14 Распространение в мире

Поиск железнодорожной колеи производится в два этапа:

- обработка изображения для выделения контура,
- применение метода Хафа для поиска прямых.

Также возможно применение фильтра Калмана для минимизации шумов анализа системы.

2.3.1. Метод Кенни и его адаптация для выделения контуров на изображении

Выделения контуров на изображении производится при помощи библиотеки OpenCV. Было проведено сравнение четырех различных способов выделения контуров:

1. Оригинальная реализация фильтра Кенни с применением функции Canny.
 - Свертка с операторами Собеля: G_x и G_y дает информацию о приближенном значении горизонтальной и вертикальной компонент вектора градиента.
 - Для каждой точки вычисляется приближенная магнитуда вектора градиента и выполняется её фильтрация по верхнему и нижнему пороговому значению.
2. Измененный метод Canny. В данной реализации отсутствует этап свертки с G_y , что позволяет избавиться от анализа горизонтальных границ. Логика двойной пороговой фильтрации сохранена, она применяется только к горизонтальной составляющей градиента.
3. Алгоритм, использующий свертку с оператором G_x и одноуровневый пороговый фильтр. Анализируются только вертикальные границы и проекция вектора градиента на ось x .
4. Свертка с операторами G_x и G_y с одноуровневой пороговой фильтрацией магнитуды градиента и дополнительным ограничением на его направление.

Алгоритмы выделения границ приведены в Приложении В.

2.3.2. Применение метода Хафа и интерпретация его результатов

Библиотека OpenCV содержит методы HoughLines (основан на оригинальном преобразовании Хафа) и HoughLinesP (вероятностное

преобразование Хафа). Второй метод производит предварительное вычисление контрольных точек на изображении и вычисляет преобразование только для них, что позволяет уменьшить количество вычислений.

Результатом работы данного метода является набор всевозможных прямых, найденных на изображении. На рисунке 15 приведен пример работы результата поиска прямых на изображении.

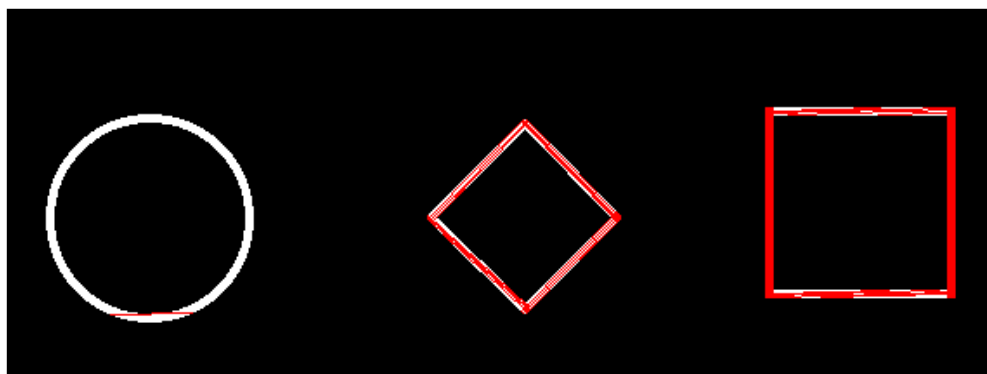


Рисунок 15 Пример выделения прямых линий на изображении. Красным цветом отмечены результаты применения преобразования Хафа.

Результаты преобразования необходимо отфильтровать по минимальной допустимой длине (предполагается, что рельс непрерывен и прямолинеен хотя бы на расстоянии большем, чем ширина колеи); выбрать только те линии, угол вектора направления которых, лежит в пределах $\pm 30^\circ$ от оси камеры; сгруппировать выделенные линии при условии, что расстояние между ними не более 5 пикселей. Дополнительно может быть проведена фильтрация при помощи определения расстояния от прямой до точки схода большинства выделенных прямых, подробно данный подход рассмотрен в работе [4].

После замены групп линий на направляющие прямые необходимо выбрать только те, что принадлежат рельсам. На данном шаге возможно ошибочное выделение:

- Контур соседних рельс, расположенных параллельно с теми, по которым движется состав;
- Внешней границы ж/д насыпи;
- Внешнего контура границы шпал;

- Дополнительных рельс в случае если на данном участке применяется 2 колеи различной ширины

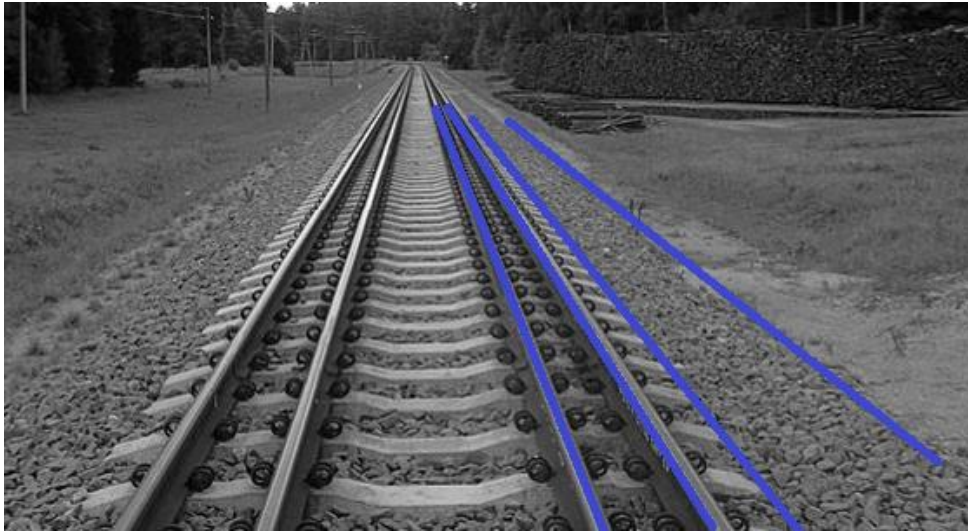


Рисунок 16 Возможные ситуации при поиске контура рельс

Пользуясь предположением о том, что ось камеры всегда расположена между рельс, даже на участках маневра, введем условие: в качестве ж/д колеи будем выделять такие 2 линии, которые:

- Проходят через точку схода большинства линий, полученных после предварительной фильтрации.
- Направление линии не выходит за ограничение: $\pm 30^\circ$ от оси камеры.
- Линии расположены по разные стороны от вертикальной линии, расположенной по центру изображения (назовем её основным направлением).
- В качестве «рельса» определяются линии, выделенные контурным анализом, наиболее близкие к основному направлению.

2.3.3. Виды ошибок при выделении рельс. Применение фильтра Калмана

Воспользуемся системой 2.2 из параграфа 1.7:

$$\begin{cases} x_{k+1} = x_k + u_k + \xi_k \\ y_k = x_k + \eta_k \end{cases}$$

Здесь

- $x_k = \begin{pmatrix} \alpha_{x1} \\ \alpha_{x2} \end{pmatrix}_k$ является двумерным вектором, первая компонента которого отвечает углу между осью камеры и правым рельсом, а вторая – между осью камеры и левым рельсом;
- $u_k = \begin{pmatrix} \alpha_{u1} \\ \alpha_{u2} \end{pmatrix}_k$ – вектор, содержащий информацию об изменении направления рельс на участке маневра;
- $y_k = \begin{pmatrix} \alpha_{y1} \\ \alpha_{y2} \end{pmatrix}_k$ – вектор наблюдения, т.е. направление рельс, полученное при помощи анализа изображения.
- k – номер кадра.

В рассматриваемой задаче под внутренней ошибкой динамической системы ξ_k можно понимать различные воздействия на камеру в процессе движения поезда. К примеру, влияние инерции камеры в момент торможения и разгона или раскачивание состава при проезде по стыкам рельс. Под ошибкой измерения η_k будем понимать погрешности при работе алгоритма выделения направления рельс.

Фильтр Калмана может быть применен в данной задаче, как способ сглаживания данных возмущений.

2.4. Построение плоскости по двум параллельным линиям и определение координат объекта интереса

Метод Виолы-Джонса позволяет определить расположение объекта интереса на плоском снимке. Однако, в задаче диагностики актуальности инфраструктуры регулирования железнодорожного движения так же возникает вопрос определения координат найденного объекта.

Для определения расположения объекта относительно камеры проведем вспомогательное построение плоскости, в которой расположены касательные прямые к рельсам. Затем спроецируем найденный объект на данную поверхность.

На рисунке 17 приведен пример проекции знака ограничения скорости движения на поверхность земли. Красным цветом отмечены направляющие рельс и вспомогательные линии, соответствующие сетке координат земной поверхности. Горизонтальная белая линия соответствует середине кадра. Зеленые линии предназначены для определения расположения знака.

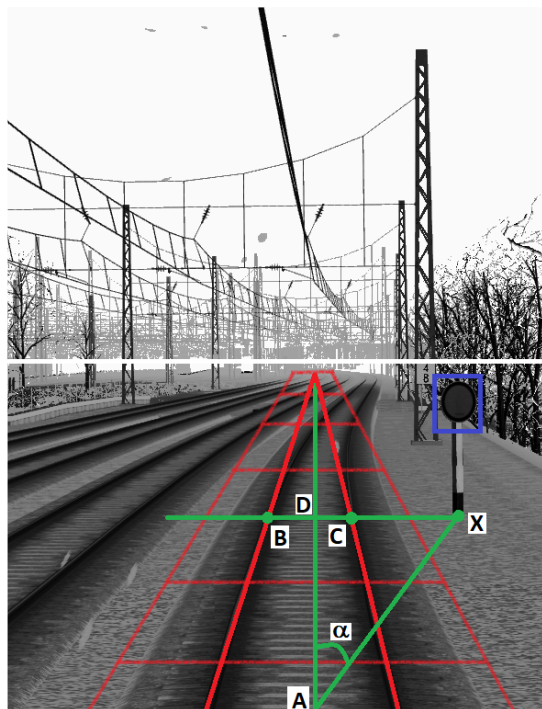


Рисунок 17 Проекция выделенного объекта на поверхность земли

Алгоритм определения координат:

1. Рассчитать высоту расположения объекта на снимке (будет рассмотрено ниже);
2. Определить точку X, соответствующую проекции объекта;
3. Построить линию, перпендикулярную оси камеры и определить точки B, C, D.
4. По расстоянию BC определить удаленность точки D согласно теории, описанной в разделе 1.4.
5. Зная AD и DX определить угол α и расстояние AX – полярные координаты объекта относительно камеры.
6. Используя абсолютные координаты камеры и относительные полярные координаты объекта определить абсолютные координаты объекта интереса.

Для вычисления высоты объекта интереса воспользуемся обобщенной теоремой Фалеса.

Теорема: Параллельные прямые, пересекающие стороны угла, отсекают от сторон угла пропорциональные отрезки.

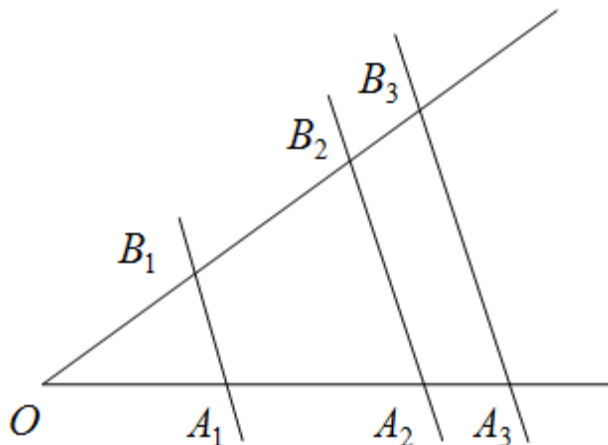


Рисунок 18 Геометрическое представление теоремы Фалеса

$$\frac{B_1B_2}{A_1A_2} = \frac{B_2B_3}{A_2A_3}$$

Рассмотрим две различные ситуации: объект интереса находится выше уровня камеры (светофор), объект интереса находится ниже уровня камеры (ж/д знак).

Рассмотрим ситуацию, при которой объект интереса находится выше уровня расположения камеры. На рисунке 19 приведен пример расчета высоты B_1D_1 объекта на снимке. Пусть B_2D_2 – реальная известная высота объекта. $АН$ – высота расположения камеры. AC_2 – линия, направленная вдоль оптической оси камеры, соответствующая горизонтальной белой линии на рисунке 17. По теореме Фалеса можно показать, что подобны следующие пары треугольников: AB_2C_2 и AB_1C_1 ; AD_2C_2 и AD_1C_1 ; AD_2C_2 и AD_1C_1 . При этом коэффициент подобия у всех пар совпадает и равен $K = \frac{B_2C_2}{B_1C_1}$, где B_1C_1 – превышение объекта на снимке относительно уровня середины снимка. $B_2C_2 = B_2D_2 - АН$ – фактическое, известное превышение объекта над уровнем камеры.

$$\text{Далее вычисляется } C_1D_1 = \frac{C_2D_2}{K} = \frac{АН}{K}, B_1D_1 = B_1C_1 + C_1D_1.$$

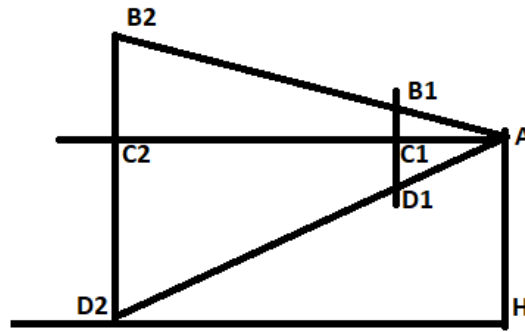


Рисунок 19 Расположение объекта интереса выше уровня камеры

В ситуации, при которой объект интереса находится ниже уровня камеры рассуждения аналогичны.

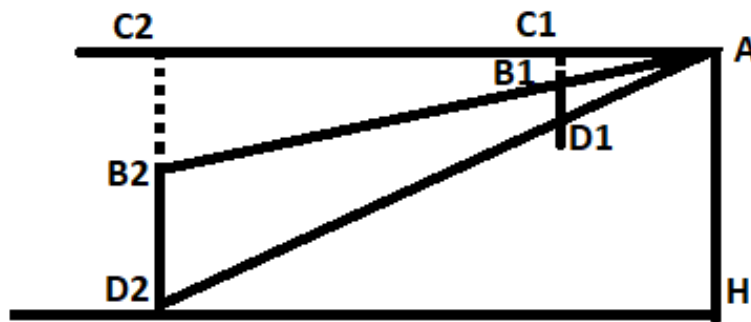


Рисунок 20 Расположение объекта интереса ниже уровня камеры

Необходимо найти отрезок B_1D_1 . Решение аналогично:

$$B_2C_2 = AH - B_2D_2$$

$$K = \frac{B_2C_2}{B_1C_1}$$

$$C_1D_1 = \frac{C_2D_2}{K} = \frac{AH}{K}$$

$$B_1D_1 = C_1D_1 - B_1C_1.$$

2.5. Выбор области интереса в соответствии с формой железнодорожной колеи

Информация, полученная при построении проекции плоского снимка на трехмерную плоскость, может быть использована при адаптивном выборе области интереса работы алгоритма.

Рассмотрим процесс поиска объекта интереса на изображении. Алгоритм Виолы-Джонса достаточно быстро обрабатывает отдельные

изображения, однако в зависимости от различных условий обучающих и тестовых изображений возможно увеличение ложных срабатываний (false positive). Можно воспользоваться информацией о расположении рельс и сократить область поиска. Так, к примеру, железнодорожные знаки ниже уровня кабины машиниста и при этом достаточно близко к колее, в связи с этим область интереса может быть уменьшена за счёт рассмотрения только нижней части изображения, также можно уменьшить ширину области интереса. На рисунке 21 приведен пример ограничения для области поиска знака (красным цветом обозначена ж/д колея, зеленым – возможные области расположения знака, синим – найденный объект интереса).

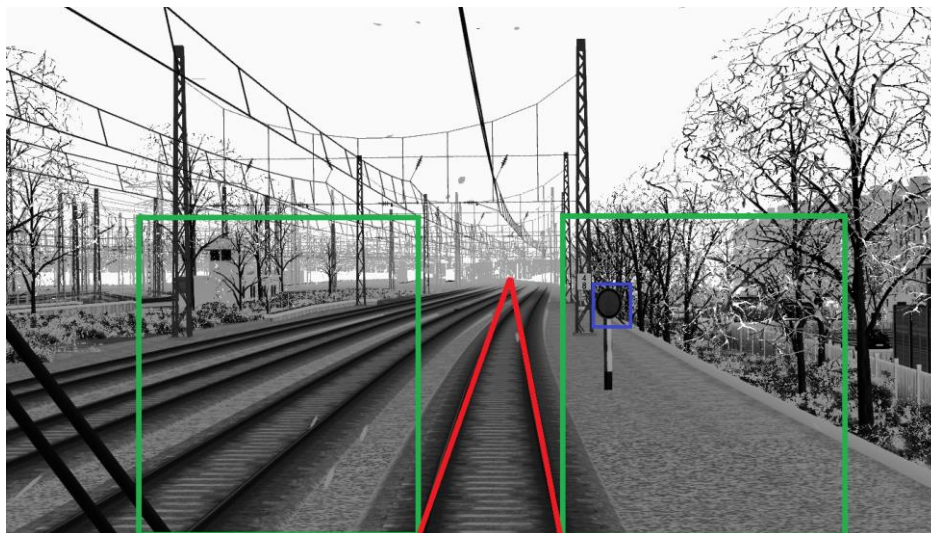


Рисунок 21 Пример ограничения области поиска ж/д знака

Светофор, как правило, располагается выше уровня кабины машиниста, так же он может быть расположен непосредственно над ж/д путями, а не только рядом с ними. Пример ограничения области поиска светофора приведен на рисунке 22.

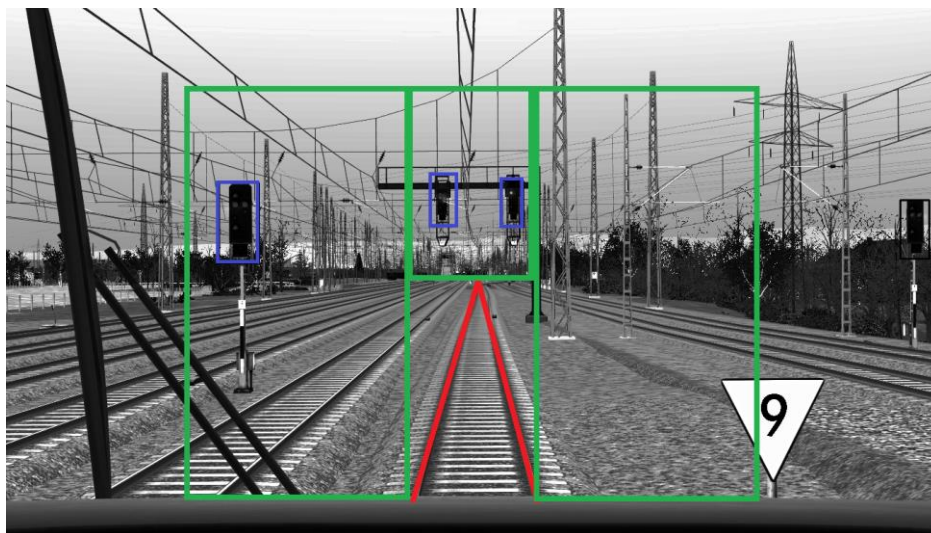
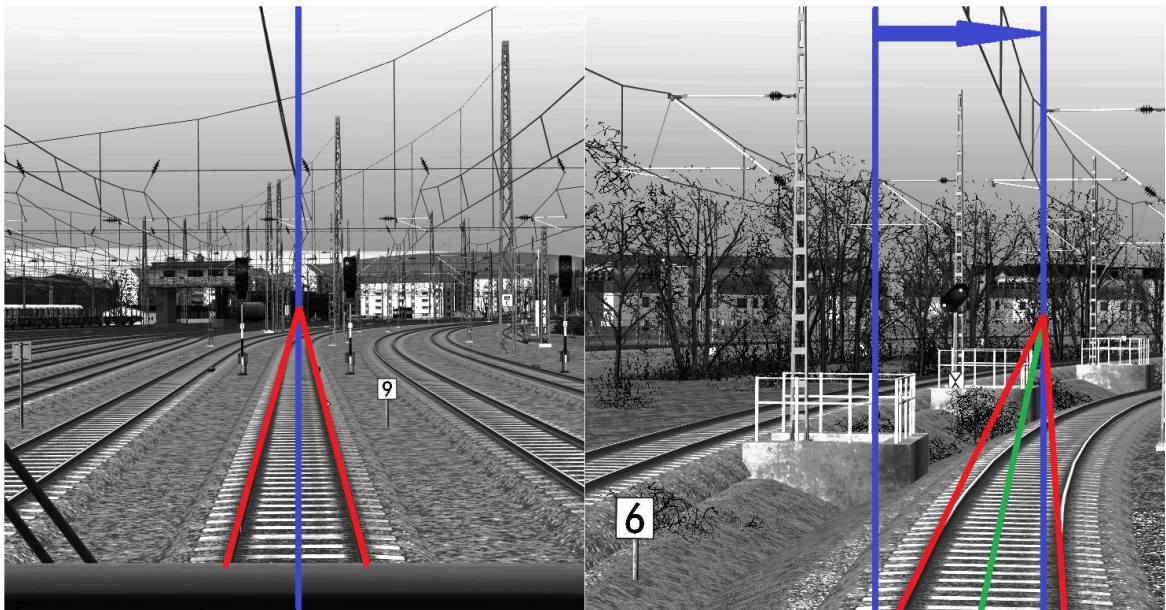


Рисунок 22 Пример ограничения области поиска ж/д светофора

В параграфе 2.3.2 мы определили термин основного направления и ввели предположение о том, что рельсы расположены по разные стороны от него. Данное предположение верно только в случае прямолинейного движения, либо при прохождении участка маневра с малым закруглением. На рисунке 23а рельсы обозначены красным цветом, а основное направление – синим. Начальное положение основного направления – центр изображения. В случае прохождения участка маневра может возникнуть ситуация, при которой оба рельса окажутся по одну сторону от основного направления (рисунок 23б, синяя линия слева). В такой ситуации имеет смысл изменить основное направления для увеличения применимости алгоритма. Новое расположение будет проходить через точку пересечения направляющих рельс, выделенных на предыдущем шаге (рисунок 23б, синяя линия справа). Максимальные границы применимости условий, описанных в параграфе 2.3.2, будут расширены до ситуации, при которой одна из направляющих будет расположена вертикально.



а

б

Рисунок 23 Смещение основного направления при прохождении участка поворота
(а – прямолинейное движение, б – участок маневра)

2.6. Вывод по главе

Описанные методы можно представить в виде следующей диаграммы (рисунок 24). Пунктирная линия отображает процесс обратной связи, т.е. влияние результатов обработки кадра на процесс анализа следующего.

Для каждого кадра:

1. Выбирается область интереса в соответствии с информацией, полученной с предыдущего кадра;
2. Дальнейшие вычисления можно проводить параллельно
 - 2.1. Поиск объекта интереса при помощи алгоритма Виолы-Джонса;
 - 2.2. Контурный анализ и преобразование Хафа для выделения рельс на изображении;
3. По результатам вычисления расположения поверхности земли на изображении выполняется:
 - 3.1. Построение области интереса для обработки следующего кадра;
 - 3.2. Вычисление расположения объектов, найденных методом Виолы-Джонса.

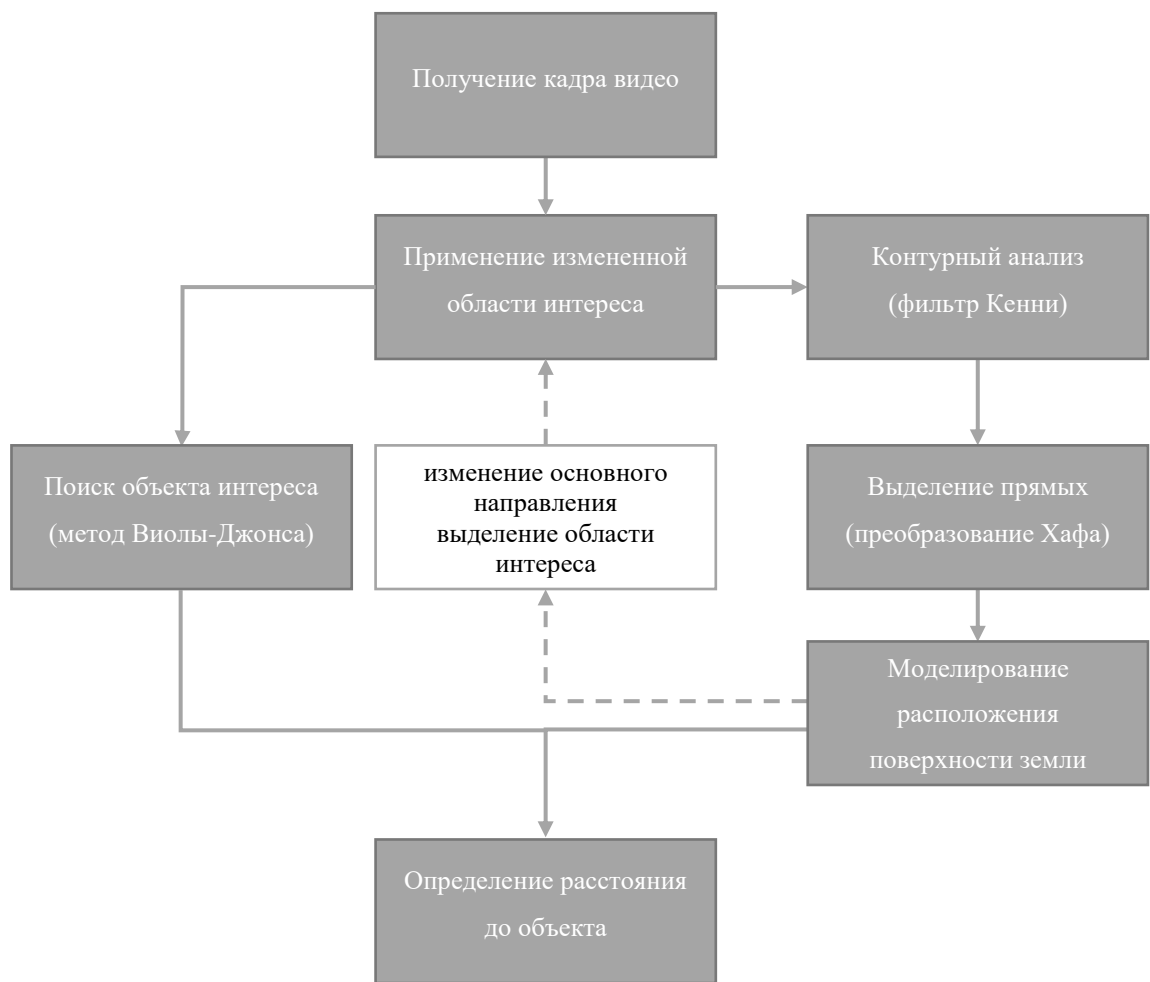


Рисунок 24 Диаграмма процесса обработки отдельного кадра видео

ГЛАВА 3. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

3.1. Контроль точности классификации

В параграфе 2.2 рассмотрен вопрос обучения каскадного классификатора Виолы-Джонса. Так как объекты интереса имеют разную форму, а метод Виолы-Джонса предназначен для выделения общих признаков, то в рамках данной работы было обучено три независимых каскадных классификатора. Для анализа качества классификации принято использовать метрики точности, полноты и F -меры. Укажем некоторые определения:

TP (true positive) — истинно-положительное решение;

TN (true negative) — истинно-отрицательное решение;

FP (false positive) — ложно-положительное решение;

FN (false negative) — ложно-отрицательное решение.

Таблица 1: Определения, применяемые при анализе качества классификации

| | | Решение эксперта | |
|-----------------|---------------|------------------|---------------|
| | | Положительное | Отрицательное |
| Решение системы | Положительное | TP | FP |
| | Отрицательное | FN | TN |

Точность классификации определяется, как:

$$Precision = \frac{TP}{TP + FP}$$

Полнота классификации определяется, как:

$$Recall = \frac{TP}{TP + FN}$$

F -мера:

$$F = 2 \frac{Precision * Recall}{Precision + Recall}$$

Набор тестовых изображений включает:

- 40 изображений (знак ограничения скорости)
- 50 изображений (светофор)
- 80 изображений (знак приближения к ж/д переезду)

Тестирование проводилось на полноразмерных изображениях (Таблица 2) и изображениях с ограниченной областью интереса (Таблица 3, см. параграф 2.5).

Таблица 2: Результаты тестирования классификатора (полное изображение)

| Объект | Кол-во изоб-ий | TP | FP | FN | Precision | Recall | F |
|----------------------------|----------------|----|----|----|-----------|--------|-------|
| Ограничение скорости | 40 | 35 | 0 | 5 | 1 | 0.875 | 0.933 |
| Приближение к ж/д переезду | 80 | 80 | 26 | 0 | 0.755 | 1 | 0.860 |
| Светофор | 50 | 50 | 11 | 0 | 0.820 | 1 | 0.901 |

Таблица 3: Результаты тестирования классификатора (ограниченная область поиска)

| Объект | Кол-во изоб-ий | TP | FP | FN | Precision | Recall | F |
|----------------------------|----------------|----|----|----|-----------|--------|-------|
| Ограничение скорости | 40 | 35 | 0 | 2 | 1 | 0.926 | 0.962 |
| Приближение к ж/д переезду | 80 | 80 | 12 | 0 | 0.870 | 1 | 0.930 |
| Светофор | 50 | 50 | 5 | 0 | 0.909 | 1 | 0.952 |

Стоит отметить, что:

- При тестировании анализа полноразмерного изображения присутствует высокий уровень FP для знака «Приближение к ж/д переезду». Ограничение области интереса позволяет существенно снизить данный показатель.
- F-мера в случае анализа ограниченной области превышает 90%.
- Т.к. основная задача системы: диагностика расположения инфраструктуры железнодорожного регулирования и сопоставление полученных результатов с цифровой картой местности, то можно утверждать, что высокие показатели полноты определения знаков и светофора говорят о том, что метод Виолы-Джонса может успешно применяться для поиска данных объектов на изображении.

Результаты можно сравнить с аналогичными исследованиями в области автомобильного транспорта. Так в статье [20] приводятся следующие результаты: Recall = 0.9703, Precision = 0.8625, F-мера = 0.9132. Сравнивая данные результаты с полученными методом Виолы-Джонса можно сделать вывод о том, что применение данного классификатора в задаче поиска и распознавания объектов железнодорожной инфраструктуры позволяет достичь высокого уровня точности и полноты результатов анализа.

3.2. Результаты сравнения методов выделения контура рельс

В параграфе 2.3.1 были рассмотрены различные вариации метода Кенни в применении к задаче выделения контура рельс.

Было проведено сравнение четырех методов:

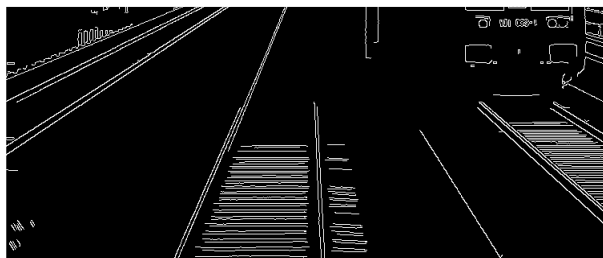
1. Оригинальная реализация фильтра Кенни с применением функции Canny из библиотеки OpenCV: Свертка с операторами G_x и G_y ; двухуровневый пороговый фильтр магнитуды градиента.
2. Измененный метод Canny: Свертка только с операторами G_x ; двухуровневый пороговый фильтр магнитуды градиента.
3. Алгоритм, использующий свертку с оператором G_x и одноуровневый пороговый фильтр магнитуды градиента.
4. Свертка с операторами G_x и G_y ; одноуровневый пороговой фильтр магнитуды градиента и дополнительным ограничением на его направление.

Методы сравнивались в трех различных ситуациях:

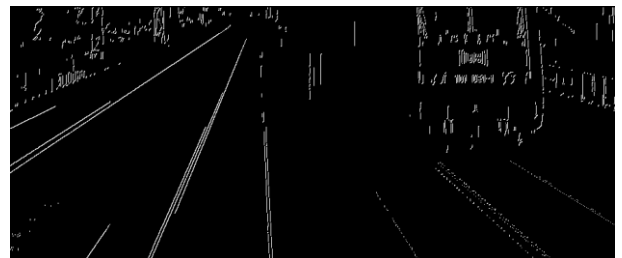
- Прямая колея без сильных отклонений и пересечений



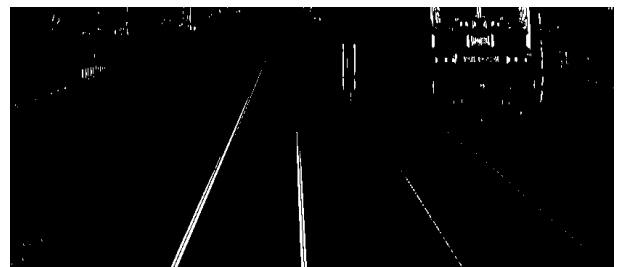
Рисунок 25 Оригинальное изображение



Метод 1



Метод 2



Метод 3

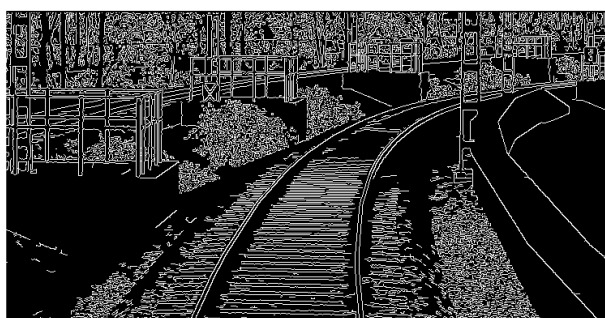
Метод 4

Рисунок 26 Сравнение методов выделения границ в случае движения по прямолинейному участку

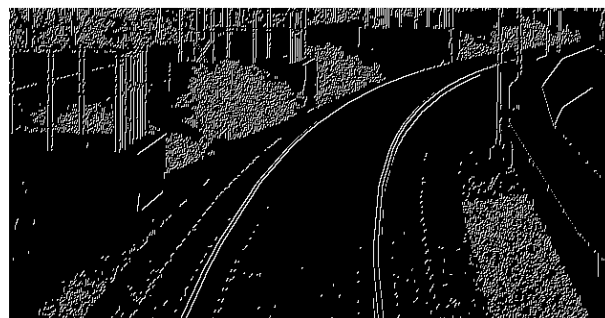
- Участок маневра



Рисунок 27 Оригинальное изображение



Метод 1



Метод 2



Метод 3



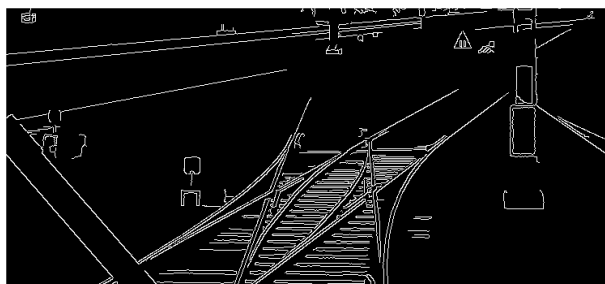
Метод 4

Рисунок 28 Сравнение методов выделения границ в случае движения по участку поворота

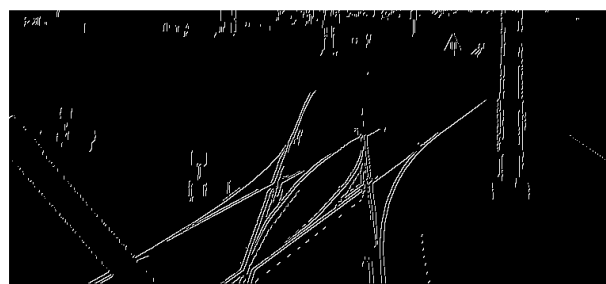
- Пересечение путей



Рисунок 29 Оригинальное изображение



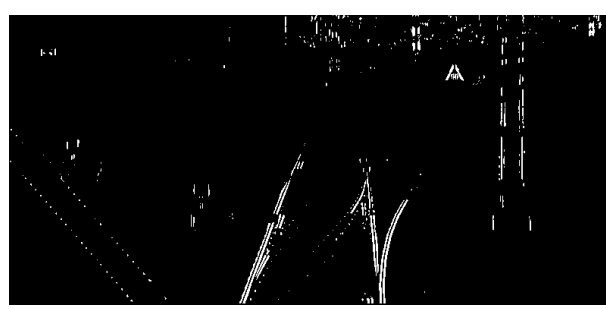
Метод 1



Метод 2



Метод 3



Метод 4

Рисунок 30 Сравнение методов выделения границ в случае движения по участку пересечения

Описание результатов сравнения методов выделения контуров:

Метод 1: Можно сделать вывод, что оригинальный метод Кенни в данной задаче дает информацию о горизонтальных и диагональных границах, которые не должны быть использованы в дальнейшем анализе.

Метод 2: Сохраняет высокую способность к выделению вертикальных границ за счет применения двухуровневого фильтра из оригинальной реализации метода Кенни.

Метод 3: Результаты сравнимы с результатами метода 2, однако часть информации о границах была потеряна по причине использования одноуровневого фильтра магнитуды градиента. Это хорошо заметно в ситуации проезда по участку пересечения рельс. В то же время, данный метод позволяет лучше справляться с различными шумами и искажениями, см. участок маневра.

Метод 4: В ситуации прямолинейного движения о параллельных путях была скрыта фильтром направления градиента. Аналогично в ситуации пересечения рельс можно увидеть выделение основного направления движения. Однако, данный фильтр приводит к потере части границ в случае проезда по участку маневра.

Из вышесказанного можно сделать вывод, что методы 3 и 4 выглядят более предпочтительными для решения данной задачи. Выбор одного основного среди них зависит от начальных предположений о наличии участков маневра.

3.3. Контроль скорости обработки одного кадра видео

В рамках работы был проведен анализ скорости обработки отдельного кадра двумя методами:

- Последовательное выполнение метода Виолы-Джонса и процедуры контурного анализа с выделением рельс
- Параллельное вычисление метода Виолы-Джонса и процедуры контурного анализа с выделением рельс

В таблице 3 (обработка полного изображения) и таблице 4 (обработка области изображения по алгоритму из параграфа 2.5) приведены результаты измерения общего времени обработки кадра и отдельных шагов его обработки. Код программы приведен в приложении А.

Таблица 3: Время обработки одного кадра видео (полное изображение). Время указано в миллисекундах.

| | | Общее время обработки кадра | Поиск объекта (Метод Виолы-Джонса) | Выделение границ (Оператор Собеля) | Детекция рельс (преобразование Хафа) |
|-------------------------|-----------|-----------------------------|------------------------------------|------------------------------------|--------------------------------------|
| Последовательный подход | Среднее | 343.94 | 226.84 | 113.31 | 3.79 |
| | Стд.откл. | 45.06 | 36.33 | 16.66 | 1.76 |
| Параллельный подход | Среднее | 240.79 | 240.71 | 161.89 | 5.67 |
| | Стд.откл. | 41.11 | 41.11 | 30.95 | 6.34 |

Таблица 4: Время обработки одного кадра видео (ограниченная область поиска). Время указано в миллисекундах.

| | | Общее время обработки кадра | Поиск объекта (Метод Виолы-Джонса) | Выделение границ (Оператор Собеля) | Детекция рельс (преобразование Хафа) |
|-------------------------|-----------|-----------------------------|------------------------------------|------------------------------------|--------------------------------------|
| Последовательный подход | Среднее | 183.18 | 71.81 | 107.10 | 4.27 |
| | Стд.откл. | 8.85 | 3.93 | 7.50 | 0.29 |
| Параллельный подход | Среднее | 131.29 | 82.94 | 126.79 | 4.40 |
| | Стд.откл. | 41.85 | 20.79 | 41.83 | 0.52 |

По результатам данного анализа можно сделать вывод о том, при параллельном подходе в обработке ограниченной области интереса достигается скорость 6-7 кадров в секунду, что является достаточным в данной задаче. Объект интереса будет присутствовать на нескольких кадрах, что увеличит шансы его обнаружения. Можно рассматривать применимость системы для обработки видео в реальном времени.

ВЫВОДЫ

Результатами данной работы являются:

1. Построена модель процесса обработки кадра видео.
2. Обучен классификатор Виолы-Джонса для поиска объекта интереса на изображении. Определена точность и полнота результатов анализа тестовых изображений. Показатель F-мера превысил 90%.
3. Проведено сравнение различных модификаций алгоритма Кенни.
4. Построен алгоритм выделения железнодорожной колеи. И определены условия его применимости.
5. Рассмотрен способ построения плоскости земли для расчета относительных координат объектов.
6. Предложен алгоритм определения расстояния до объекта интереса.
7. Описан процесс уменьшения области поиска знаков и светофоров при анализе информации об относительном расположении объектов в трехмерном пространстве
8. Скорость полной обработки одного кадра видео находится на уровне 6-7 кадров в секунду, что позволяет рассмотреть возможность применения данной системы при обработке видео в реальном времени.

В качестве задач дальнейшего исследования можно рассмотреть:

1. Сравнение эффективности метода Виолы-Джонса и сверточных нейронных сетей в задаче детекции средств регулирования на железной дороге.
2. Реализовать применение фильтра Калмана в задаче выделения рельс.
3. Определить эффективность алгоритма, построенного при помощи симулятора, при работе в реальных условиях.

Список литературы

1. Абламейко С.В., Лагуновский Д.М. Обработка изображений: технология, методы, применение. Учебное пособие. М.: Амалфея, 2000. 18-44 стр.
2. Браммер К., Зиффлинг Г. Фильтр Калмана-Бьюси. Детерминированное наблюдение и стохастическая фильтрация. 1982: Наука. Главная редакция физико-математической литературы.
3. Власенко С. Системы АЛС для высокоскоростного сообщения // Автоматика связь информатика. 011. No. 3. P. 39.
4. Иванов Ю.А. Разработка локомотивной системы технического зрения. Диссертация на соискание учёной степени кандидата технических наук. М.: ФГБОУ ВПО "Московский авиационный институт" (МАИ), 2014.
5. Инструкция по сигнализации на железнодорожном транспорте Российской Федерации. 2012.
6. Рязанов С.Н. Транспортная безопасность объектов железнодорожной инфраструктуры // Автоматика, связь, информатика. 2011. Вып. 6. стр. 23-25.
7. Труфанов М.И., С.В. П. Способ обнаружения препятствий перед транспортным средством с использованием бинокулярной системы технического зрения.
8. Усилин С.А. Алгоритмическое развитие Виола-Джонсовских детекторов для решения прикладных задач распознавания изображений. М.. 2017.
9. Ballard D.H. Generalizing the Hough transform to detect arbitrary shapes // Pattern Recognition 13, 2, April 1981.
10. Bileschi S.M. StreetScenes: Towards scene understanding in still images. PhD thesis. Citeseer. 2006.
11. Canny J. Computational Approach To Edge Detection, IEEE Trans // Pattern Analysis and Machine Intelligence, No. 8(6), 1986. pp. 679–698.
12. Filipowicz A. Driving School II. Video Games for Autonomous Driving. 2016.
13. Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. // Conference on Computer Vision and Pattern Recognition (CVPR). 2012.
14. Hough P. Method and means for recognizing complex patterns. U.S. Patent 3,069,654. December 18, 1962.

15. Kalman R.E. A new approach to linear filtering and prediction problems // Journal of Basic Engineering 82 (1), 1960. pp. 35-45.
16. Karakose M., Yaman O., Baygin M., Murat K., Akin E. A New Computer Vision Based Method for Rail Track Detection and Fault Diagnosis in Railways // International Journal of Mechanical Engineering and Robotics Research, Vol. 6, No. 1, January 2017. pp. 22-27.
17. Karla Brkie. An overview of traffic sign detection methods. Zagreb: Department of Electronics, Microelectronics, Computer and Intelligent Systems. Faculty of Electrical Engineering and Computing.
18. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989.
19. Lienhart R., Kuranov A., Pisarevsky V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection With an Extended Set of Haar-like Features, – Intel Technical Report. 2002.
20. Nadra Ben Romdhane, Hazar Mliki, Mohamed Hammami. A new approach to traffic sign recognition through primary visual characteristics // Computer and Information Science (ICIS) 2016 IEEE/ACIS 15th International Conference on, 2016. pp. 1-6.
21. Puttemans S. OpenCV Tutorials. Object Detection. Cascade Classifier Training. https://docs.opencv.org/3.3.0/dc/d88/tutorial_traincascade.html.
22. Ruta A., Li Y., Liu X. Real-time traffic sign recognition from video by class-specific discriminative features // Pattern recognition, Vol. 43, No. 1, 2010. pp. 416–430.
23. Sobel I., Feldman G. A 3x3 Isotropic Gradient Operator for Image Processing (неопубл.), 1968.
24. Viola P., Jones M. Robust real-time object detection // International Journal of Computer Vision, 2001.
25. NVidia Driver Assistance. <http://www.nvidia.com/object/advanced-driver-assistance-systems.html>.
26. Train Simulator http://store.steampowered.com/app/24010/Train_Simulator/
27. Vector. https://vector.com/vi_adas_en.html.

Метод для обработки одного кадра видео

```

/// <summary>
///   Process single video frame.
/// </summary>
private void ProcessFrame(Image<Bgr, byte> initialImage)
{
    DetectedObjects objects = DetectObjects(initialImage);

    Image<Gray, byte> edgeImage = BlurEdges(DetectEdges(initialImage));

    DetectedRails rails = DetectRails(edgeImage);

    MeasureDistance(rails, objects);

    Display(initialImage, objects, edgeImage, rails);
}

```

```

/// <summary>
///   Process single video frame in parallel.
/// </summary>
private void ProcessFrameInParallel(Image<Bgr, byte> initialImage)
{
    DetectedObjects objects = null;
    Image<Gray, byte> edgeImage = null;
    DetectedRails rails = null;

    Task task1 = Task.Factory.StartNew(() =>
    {
        objects = DetectObjects(initialImage);
    });

    Task task2 = Task.Factory.StartNew(() =>
    {
        edgeImage = BlurEdges(DetectEdges(initialImage));
        rails = DetectRails(edgeImage);
    });

    Task.WhenAll(task1, task2).Wait();

    MeasureDistance(rails, objects);

    Display(initialImage, objects, edgeImage, rails);
}

```


Метод для поиска объектов интереса

```
private readonly CascadeClassifier crossCascadeClassifier =
    new CascadeClassifier("road-cross-cascade.xml");
private readonly CascadeClassifier lightsCascadeClassifier =
    new CascadeClassifier("lights-cascade.xml");
private readonly CascadeClassifier limitCascadeClassifier =
    new CascadeClassifier("speed-limit-cascade.xml");

/// <summary>
/// Detect objects by Viola-Jones algorithm.
/// </summary>
private DetectedObjects DetectObjects(Image<Bgr, byte> initialImage)
{
    Image<Gray, byte> gray = initialImage.Convert<Gray, byte>();
    // apply ROI to detect light. Information from the previous iteration results
    Memory.ApplyLightsROI(gray);
    List<Rectangle> lights = lightsCascadeClassifier
        .DetectMultiScale(gray, 1.1D, 10, new Size(10, 20), new Size(150, 150))
        .ToList();
    // apply ROI to detect sign. Information from the previous iteration results
    Memory.ApplySignROI(gray);
    List<Rectangle> limits = limitCascadeClassifier
        .DetectMultiScale(gray, 1.1D, 10, new Size(10, 10), new Size(150, 150))
        .ToList();

    List<Rectangle> cross = crossCascadeClassifier
        .DetectMultiScale(gray, 1.1D, 10, new Size(10, 20), new Size(150, 150))
        .ToList();

    return new DetectedObjects
    {
        Lights = lights,
        Limits = limits,
        Cross = cross
    };
}
```

```

/// <summary>
/// Selector of edge detection method.
/// </summary>
private Image<Gray, byte> DetectEdges(Image<Bgr, byte> initialImage)
{
    const double thresholdLow = 150;
    const double thresholdHigh = 250;
    const int detectorMethodType = 3;
    Image<Gray, byte> gray = initialImage.Convert<Gray, byte>();

    switch (detectorMethodType)
    {
        // Canny original (two dimension, two level threshold)
        case 1: return Canny(gray, thresholdLow, thresholdHigh);
        // Canny modified (one dimension, two level threshold)
        case 2: return CannyModified(gray, thresholdLow, thresholdHigh);
        // Sobel operator (one dimension, one level threshold)
        case 3: return Sobel(gray, thresholdLow);
        // Sobel operator (two dimension, edge direction filter, one level threshold)
        case 4: return SobelWithDirection(gray, thresholdLow);
        default: return null;
    }
}

/// <summary>
/// Method 1: Original Canny filter implementation.
/// </summary>
private Image<Gray, byte> Canny(Image<Gray, byte> gray, double thresholdLow, double
thresholdHigh)
{
    return gray.Canny(thresholdLow, thresholdHigh);
}

/// <summary>
/// Method 2: Modified Canny filter.
/// </summary>
private Image<Gray, byte> CannyModified(Image<Gray, byte> gray, double thresholdLow,
double thresholdHigh)
{
    //call to C++ library
    return cvCannyModified(gray, thresholdLow, thresholdHigh);
}

/// <summary>
/// Method 3: Sobel operator (Gx).
/// </summary>
private Image<Gray, byte> Sobel(Image<Gray, byte> gray, double thresholdLow)
{
    return gray
        .SmoothGaussian(3)
        .Sobel(1, 0, 3) // calculate horizontal gradient magnitude (vertical edges)
        .InRange(new Gray(thresholdLow), new Gray(255))
        .Convert<Gray, byte>();
}

```

```

/// <summary>
///   Method 4: Sobel operators (Gx, Gy) and direction threshold.
/// </summary>
private Image<Gray, byte> SobelWithDirection(Image<Gray, byte> gray, double
thresholdLow)
{
    const double direction = 1.5;

    var verticalEdges = gray
        .SmoothGaussian(3)
        .Sobel(1, 0, 3) // calculate horizontal gradient magnitude (vertical edges)
        .InRange(new Gray(thresholdLow), new Gray(255));

    var horizontalEdges = gray
        .SmoothGaussian(3)
        .Sobel(0, 1, 3) // calculate vertical gradient magnitude (horizontal edges)
        .InRange(new Gray(thresholdLow), new Gray(255));

    var result = new Image<Gray, byte>(gray.Size);
    for (int i = 2; i < gray.Rows - 2; i++)
    {
        for (int j = 2; j < gray.Cols - 2; j++)
        {
            if (Math.Abs(verticalEdges[i, j].Intensity) >
                direction * Math.Abs(horizontalEdges[i, j].Intensity))
            {
                result[i, j] = white;
            }
        }
    }
    return result;
}

```